

# C++ Data Types

In this tutorial, we will learn about basic data types such as int, float, char, etc. in C++ programming with the help of examples.

In C++, data types are declarations for variables. This determines the type and size of data associated with variables. For example,

```
int age = 13;
```

Here, *age* is a variable of type `int`. Meaning, the variable can only store integers of either 2 or 4 bytes.

---

## C++ Fundamental Data Types

The table below shows the fundamental data types, their meaning, and their sizes (in bytes):

| Data Type            | Meaning               | Size (in Bytes) |
|----------------------|-----------------------|-----------------|
| <code>int</code>     | Integer               | 2 or 4          |
| <code>float</code>   | Floating-point        | 4               |
| <code>double</code>  | Double Floating-point | 8               |
| <code>char</code>    | Character             | 1               |
| <code>wchar_t</code> | Wide Character        | 2               |
| <code>bool</code>    | Boolean               | 1               |
| <code>void</code>    | Empty                 | 0               |

Now, let us discuss these fundamental data types in more detail.

---

### 1. C++ int

- The `int` keyword is used to indicate integers.

- Its size is usually 4 bytes. Meaning, it can store values from - **2147483648 to 2147483647**.
- For example,

```
int salary = 85000;
```

## 2. C++ float and double

- `float` and `double` are used to store floating-point numbers (decimals and exponentials).
- The size of `float` is 4 bytes and the size of `double` is 8 bytes. Hence, `double` has two times the precision of `float`. To learn more, visit [C++ float and double](#).
- For example,

```
float area = 64.74;  
double volume = 134.64534;
```

As mentioned above, these two data types are also used for exponentials. For example,

```
double distance = 45E12    // 45E12 is equal to 45*10^12
```

## 3. C++ char

- Keyword `char` is used for characters.
- Its size is 1 byte.
- Characters in C++ are enclosed inside single quotes `' '`.
- For example,

```
char test = 'h';
```

**Note:** In C++, an integer value is stored in a `char` variable rather than the character itself. To learn more, visit [C++ characters](#).

## 4. C++ wchar\_t

- Wide character `wchar_t` is similar to the `char` data type, except its size is 2 bytes instead of 1.
- It is used to represent characters that require more memory to represent them than a single `char`.
- For example,

```
wchar_t test = L'א' // storing Hebrew character;
```

Notice the letter L before the quotation marks.

**Note:** There are also two other fixed-size character types `char16_t` and `char32_t` introduced in C++11.

---

## 5. C++ bool

- The `bool` data type has one of two possible values: `true` or `false`.
- Booleans are used in conditional statements and loops (which we will learn in later chapters).
- For example,

```
bool cond = false;
```

[https://www.youtube.com/embed/zB9RI8\\_wExo](https://www.youtube.com/embed/zB9RI8_wExo)

---

## 6. C++ void

- The `void` keyword indicates an absence of data. It means "nothing" or "no value".
- We will use void when we learn about functions and pointers.

**Note:** We cannot declare variables of the `void` type.

---

## Putting it all together

Run the following repl to see its results. Looking at the `main.cpp` file, a range of variables of different data types are declared at the top.

`cout` is used with the `+` operator to output the sum of two integers. `cout` is also used to display a message that includes the contents of the string variables, and lastly an if statement is used to present a message based on which letter is stored in the char variable.

Try doing each of the following, running the between each change:

1. Change the string variables to reflect your name
2. Change the values of the integers
3. Change the operator used in the output to `*`
4. Change the letter stored in the char variable to a grade you would like to receive
5. Change the integer variables to float, and assign decimal values.

<https://replit.com/@ChesterWhitwell/C-variables-and-operators?lite=true>

---

## C++ Type Modifiers

We can further modify some of the fundamental data types by using type modifiers. There are 4 type modifiers in C++. They are:

1. `signed`
2. `unsigned`
3. `short`
4. `long`

We can modify the following data types with the above modifiers:

- `int`
- `double`

- char

# C++ Modified Data Types List

| Data Type          | Size (in Bytes) | Meaning   |
|--------------------|-----------------|---|
| signed int         | 4               | used for integers (equivalent to <code>int</code> )   |
| unsigned int       | 4               | can only store positive integers  |
| short              | 2               | used for small integers (range <b>-32768 to 32767</b> )   |
| unsigned short     | 2               | used for small positive integers (range <b>0 to 65,535</b> )                                    |
| long               | at least 4      | used for large integers (equivalent to <code>long int</code> )                                  |
| unsigned long      | 4               | used for large positive integers or 0 (equivalent to <code>unsigned long int</code> )           |
| long long          | 8               | used for very large integers (equivalent to <code>long long int</code> ).                       |
| unsigned long long | 8               | used for very large positive integers or 0 (equivalent to <code>unsigned long long int</code> ) |
| long double        | 12              | used for large floating-point numbers   |
| signed char        | 1               | used for characters (guaranteed range <b>-127 to 127</b> )                                      |
| unsigned char      | 1               | used for characters (range <b>0 to 255</b> )  |

Let's see a few examples.

```
long b = 4523232;
long int c = 2345342;
long double d = 233434.56343;
short d = 3434233; // Error! out of range
unsigned int a = -5;    // Error! can only store positive numbers or 0
```

## Derived Data Types

Data types that are derived from fundamental data types are derived types. For example: arrays, pointers, function types, structures, etc.

We will learn about these derived data types in later tutorials.

---

Revision #6

Created 3 May 2022 02:13:27 by Chester Whitwell

Updated 3 May 2022 02:41:56 by Chester Whitwell